Event-driven simulation of benchmarks 1 & 2

The files described here contain C++ code for simulating the benchmarks 1 and 2 described in the following paper:
Simulation of networks of spiking neurons: A review of tools and strategies (2006). Brette, Rudolph, Carnevale, Hines, Beeman, Bower, Diesmann, Goodman, Harris, Zirpe, Natschläger, Pecevski, Ermentrout, Djurfeldt, Lansner, Rochel, Vibert, Alvarez, Muller, Davison, El Boustani and Destexhe. Journal of Computational Neuroscience.

These benchmarks are random networks of integrate-and-fire neurons, with either conductance-based synapses (benchmark 1) or current-based synapses (benchmark 2). The files described here are an exact event-driven implementation of these benchmarks. The implementation was programmed by Benjamin Cohen, using algorithms by Romain Brette.

Benchmark 1 (conductance-based synapses):

The implementation of the first benchmark relies on the algorithm described in the following paper:
Exact simulation of integrate-and-fire models with synaptic conductances. Brette, R. (2006). Neural Computation 18(8): 2004-2027.

The benchmark was slightly modified by imposing $\tau_e = \tau_i = 5$ ms. This is an intrinsic limitation of the method described in the paper mentioned above. Also, instead of initially stimulating the network, the state variables of the neurons are initialized randomly, so that the network is in a stable active state.

The archive contains the following files:

- conf.h: all parameters of the simulation.
- main.cpp: runs the simulation and writes the output in a file as a list of spikes produced by the network (two numbers per line: index of the neuron, spike time).
- spike.h: defines a class for spike events.
- dclist.*: a double-chained list of spikes.
- Ncq.*: a calendar queue for spikes.
- neuron.*: class with the basic event-driven operations for a neuron.
- network.*: class holding the neurons and the event queue.
- annexe.*: model-specific functions defined in the paper mentioned above.

Benchmark 2 (current-based synapses):

The implementation of the second benchmark relies on the algorithm described in the following paper:
Exact simulation of integrate-and-fire models with exponential currents. Brette, R. (2006). (Being revised).

Note that this implementation is slow because it uses non-optimal polynomial routines.

The archive contains the following files:

- conf.h: all parameters of the simulation.
- main.cpp: runs the simulation and writes the output in a file as a list of spikes produced by the network (two numbers per line: index of the neuron, spike time).
- spike.h: defines a class for spike events.

- dclist.*: a double-chained list of spikes.
- Ncq.*: a calendar queue for spikes.
- neuron.*: class with the basic event-driven operations for a neuron.
- network.*: class holding the neurons and the event queue.
- polynomial.*: polynomial operations